IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s):	Om P. Agrawal; Bai Nguyen; Kuang Chi; Brad Sharpe-Geisler; Giap Tran		
Assignee:	Lattice Semiconductor Corporation		
Title:	Scalable Serializer-Deserializer Architecture And Programmable Interface		
Serial No.:	10/619,645	Filing Date:	07/14/2003
Examiner:	Jason Crawford	Group Art Unit:	2819
Docket No.:	M-15135 US	Confirmation No.:	9559

Irvine, California
March 20, 2006

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

EXHIBIT B FOR

37 C.F.R. 1.131 DECLARATION

LAW OFFICES OF
MACPHERSON KWOK
CHEN & HEID LLP

2402 Michelson Drive
SUITE 210
Irvine, CA 92612
(949) 752-7040
FAX (949) 752-7049

Lattice Semiconductor Corporation

ispXPGA™ Family

March 2003 — Preliminary Data Sheet

- **Non-volatile, Infinitely Reconfigurable**
  - Instant-on - Powers up in microseconds via on-chip E²CMOS® based memory
  - No external configuration memory
  - Excellent design security, no bit stream to intercept
  - Reconfigure SRAM based logic in milliseconds
- **High Logic Density for System-level Integration**
  - 139K to 1.25M system gates
  - 160 to 496 I/O
  - 1.8V, 2.5V, and 3.3V V_{CC} operation
  - Up to 414Kb sysMEM™ embedded memory
- **High Performance Programmable Function Unit (PFU)**
  - Four LUT-4 per PFU supports wide and narrow functions
  - Dual flip-flops per LUT-4 for extensive pipelining
  - Dedicated logic for adders, multipliers, multiplexers, and counters
- **Variable-Length Interconnect Routing Technology**
  - Optimum speed, power, and flexibility for logic interconnections
- **Flexible Memory Resources**
  - Multiple sysMEM Embedded RAM Blocks
    - Single port, Dual port, and FIFO operation
  - 64-bit distributed memory in each PFU
    - Single port, Double port, FIFO, and Shift Register operation
- **Eight sysCLOCK™ Phase Locked Loops (PLLs) for Clock Management**
  - True PLL technology
  - 10MHz to 320MHz operation
  - Clock multiplication and division
  - Phase adjustment
  - Shift clocks in 250ps steps
- **sysIO™ for High System Performance**
  - High speed memory support through SSTL and HSTL
  - Advanced buses supported through PCI, GTL+, LVDS, BLVDS, and LVPECL
  - Standard logic supported through LVTTL, LVCMOS 3.3, 2.5, and 1.8
  - Programmable drive strength for series termination
  - Programmable bus maintenance
- **sysHSI™ Capability for Ultra Fast Serial Communications**
  - Up to 850Mbps performance
  - Up to 20 channels per device
  - Built in Clock Data Recovery (CDR) and Serialization and De-serialization (SERDES)
- **Flexible Programming, Reconfiguration, and Testing**
  - IEEE 1532 and 1149.1 compliant
  - Microprocessor configuration interface
  - Program E²CMOS while operating from SRAM

Table 1. ispXPGA Family Selection Guide

	ispXPGA 125	ispXPGA 200	ispXPGA 500	ispXPGA 1200
System Gates	139K	210K	476K	1.25M
PFUs	484	676	1764	3844
LUT-4s	1936	2704	7056	15376
Logic FFs	3.8K	5.4K	14.1K	30.7K
sysMEM Memory	92K	111K	184K	414K
Distributed Memory	30K	43K	112K	246K
EBR	20	24	40	90
sysHSI Channels	4	8	12	20
User I/O	160/176	160/208	336	496
Packaging	256 fpBGA 516 fpBGA[1]	256 fpBGA 516 fpBGA[1]	516 fpBGA[1] 900 fpBGA	680 fpSBGA[1] 900 fpBGA

1. Thermally enhanced package.

Note: LFX1200B/C is preliminary, LFX125/200/500B/C information is advanced.

ispXPGA Family Overview

The ispXPGA family of devices allows the creation of high-performance logic designs that are both non-volatile and infinitely re-programmable. Other FPGA solutions force a compromise being either re-programmable or non-volatile. This family couples this capability with a mainstream architecture containing the features required for today's system-level design.

Electrically Erasable CMOS (E^2CMOS) memory cells provide the ispXPGA family with non-volatile capability. These allow logic to be functional microseconds after power is applied, allowing easy interfacing in many applications. This capability also means that expensive external configuration memories are not required and that designs can be secured from unauthorized read back. Internal SRAM cells allow the device to be infinitely reconfigured if desired. Both the SRAM and E^2CMOS cells can be programmed and verified through the IEEE 1532 industry standard. Additionally, the SRAM cells can be configured and read-back through the sysCONFIG™ peripheral port.

The family spans the density and I/O range required for the majority of today's logic designs, 139K to 1.25M system gates and 160 to 496 I/O. The devices are available for operation from 1.8V, 2.5V, and 3.3V power supplies, providing easy integration into the overall system.

The system-level needs of designers are met through the incorporation of sysMEM dual-port memory blocks, sysIO advanced I/O support, and sysCLOCK Phase Locked Loops (PLLs). High-speed serial communications are supported through multiple sysHSI blocks, which provide clock data recovery (CDR) and serialization/de-serialization (SERDES).

The ispLEVER™ design tool from Lattice allows designers easy implementation of designs using the ispXPGA product. Synthesis library support is available for the major logic synthesis tools. The ispLEVER tool takes the output from these common synthesis packages and place and routes the design in the ispXPGA product. The tool allows floor planning and the management of other constraints within the device. The tool also provides outputs to common timing analysis tools for timing analysis.

To increase designer productivity, Lattice provides a variety of pre-designed modules referred to as IP cores for the ispXPGA product. These IP cores allow designers to concentrate on the unique portions of their design while using pre-designed block to implement standard functions such as bus-interfaces, standard communication-interfaces, and memory-controllers.

Through the use of advanced technology and innovative architecture the ispXPGA FPGA devices provide designers with excellent speed performance. Although design dependent, many typical designs can run at over 150MHz. Certain designs can run at over 300MHz. Table 2 details the performance of several building blocks commonly used by logic designers.

Table 2. ispXPGA Speed Performance for Typical Building Blocks

Function	Performance
8:1 Asynch MUX	150 MHz
1:32 Asynch Demultiplexer	125 MHz
8 x 8 2-LL Piped Multiplier	225 MHz
32-bit Up/Down Counter	290 MHz
32-bit Shift Register	360 MHz

Architecture Overview

The IspXPGA architecture is a symmetrical architecture consisting of an array of Programmable Function Units (PFUs) enclosed by Input Output Groups (PICs) with columns of sysMEM Embedded Block RAMs (EBRs) distributed throughout the array. Figure 1 illustrates the IspXPGA architecture. Each PIC has two corresponding sysIO blocks, each of which includes one input and output buffer. On two sides of the device, between the PICs and the sysIO blocks, there are sysHSI High-Speed Interface blocks. The symmetrical architecture allows designers to easily implement their designs, since any logic function can be placed in any section of the device.

The PFUs contain the basic building blocks to create logic, memory, arithmetic, and register functions. They are optimized for speed and flexibility allowing complex designs to be implemented quickly and efficiently.

The PICs interface the PFUs and EBRs to the external pins of the device. They allow the signals to be registered quickly to minimize setup times for high-speed designs. They also allow connections directly to the different logic elements for fast access to combinatorial functions.

The sysMEM EBRs are large, fast memory elements that can be configured as RAM, ROM, FIFO, and other storage types. They are designed to facilitate both single and dual-port memory for high-speed applications.

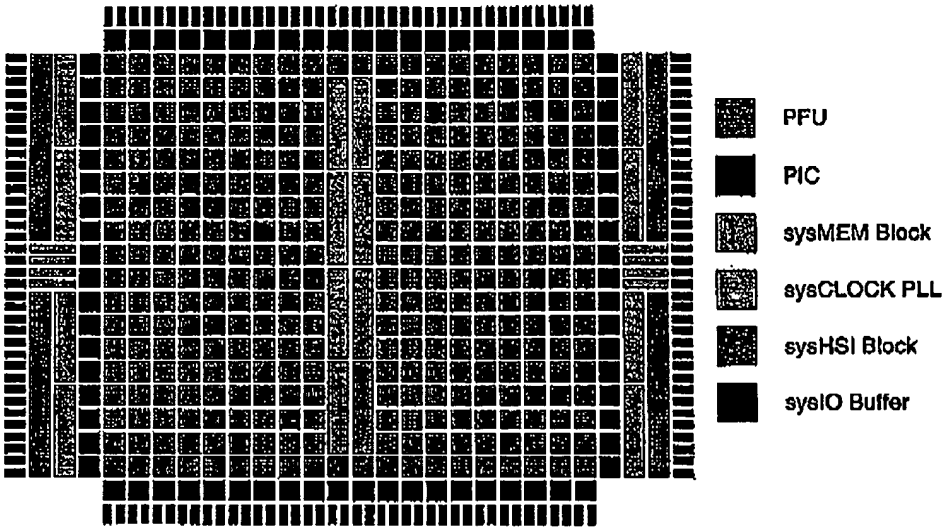
These three components of the architecture are interconnected via a high-speed, flexible routing array. The routing array consists of Variable Length Interconnect (VLI) lines between the PICs, PFUs, and EBRs. There is additional routing available to the PFU for feedback and direct routing of signals to adjacent PFUs or PICs.

The sysIO blocks consist of configurable input and output buffers connected directly to the PICs. These buffers can be configured to interface with 16 different I/O standards. This allows IspXPGA to interface with other devices without the need for external transceivers.

The sysHSI blocks provide the necessary components to allow the IspXPGA device to transfer data at up to 850Mbps using the LVDS standard. These components include serializing, de-serializing, and clock data recovery (CDR) logic.

The sysCLOCK blocks provide clock multiplication/division, clock distribution, delay compensation, and increased performance through the use of PLL circuitry that manipulates the global clocks. There is one sysCLOCK block for each global clock tree in the device.

Figure 1. IspXPGA Block Diagram



Programmable Function Unit

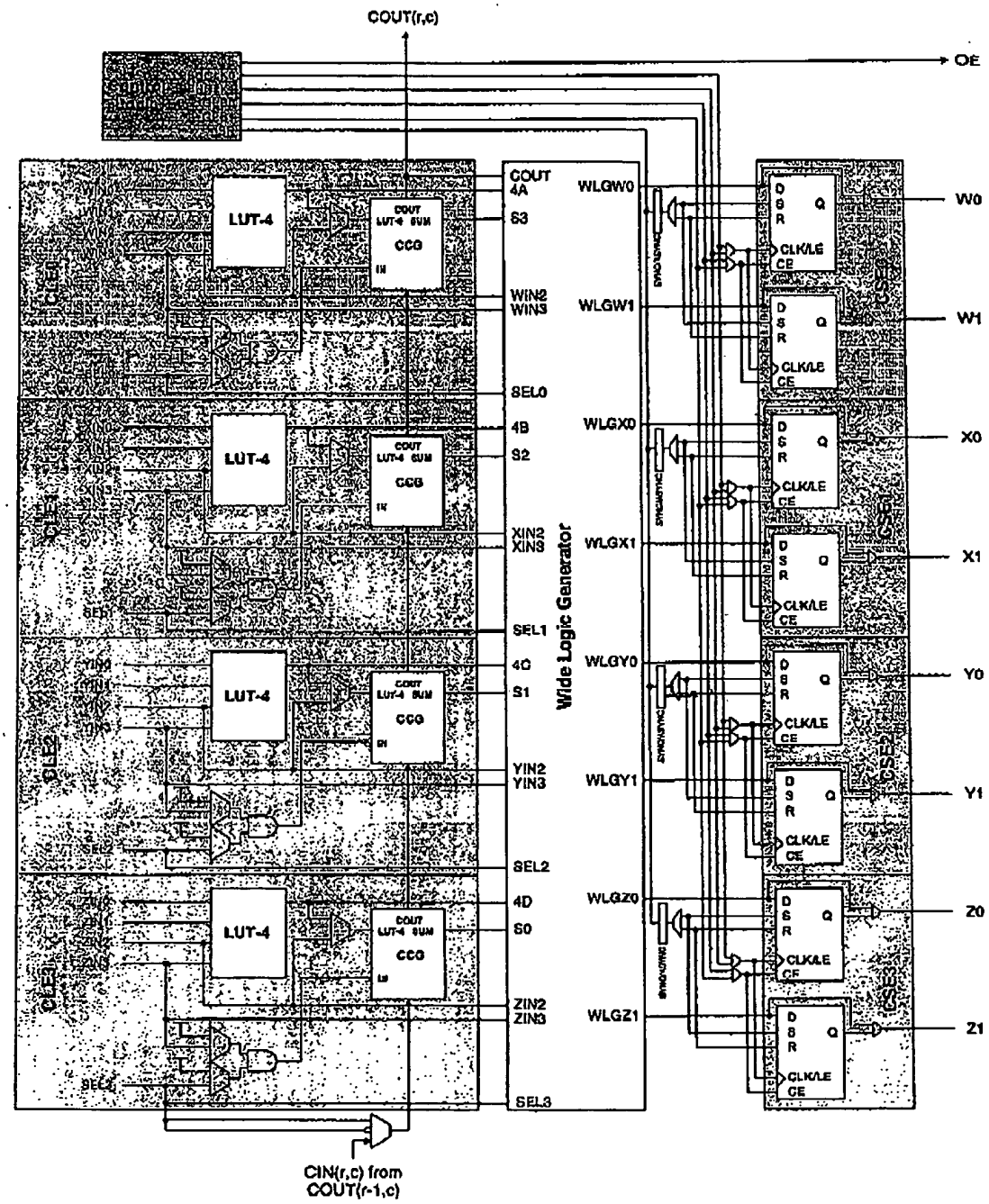
The Programmable Function Unit (PFU) is the basic building block of the IspXPGA architecture. The PFUs are arranged in rows and columns in the device with PFU (1,1) referring to (row 1, column 1). Each PFU consists of four Configurable Logic Elements (CLEs), four Configurable Sequential Elements (CSEs), and a Wide Logic Generator (WLG). By utilizing these components, the PFU can implement a variety of functions. Table 3 lists some of the function capabilities of the PFU.

There are 57 inputs to each PFU and nine outputs. The PFU uses 20 inputs for logic, and 37 inputs drive the control logic from which six control signals are derived for the PFU.

Table 3. Function Capability of IspXPGA PFU

Function	Capability
Look-up table	LUT-4, LUT-5, LUT-6
Wide logic functions	Up to 20 input logic functions
Multiplexing	2:1, 4:1, 8:1
Arithmetic logic	Dedicated carry chain and booth multiplication logic
Single-port RAM	16X1, 16X2, 16X4, 32X1, 32X2, 64X1
Double-port RAM	16X1, 16X2, 32X1
Shift register	8-bit shift registers (up to 32-bit shift capability)

Figure 2. IspXPGA PFU



Configurable Logic Element

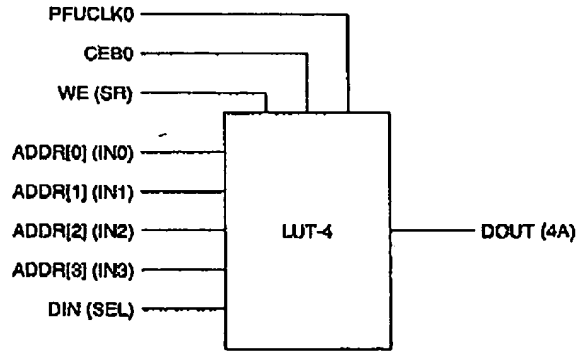
The CLE is made up of a four-input Look-up Table (LUT-4), a Carry Chain Generator (CCG), and a two-input AND gate. The LUT-4 creates various combinatorial and memory elements, the CCG creates a single one-bit full adder, and the two-input AND gate can expand the CCG to incorporate Booth Multiplier capability by feeding the output of the AND gate to one of the inputs of the CCG.

Of the five inputs that feed each CLE, two are dedicated inputs into each LUT-4 and the remaining three take on varying functionality. The third and fourth inputs can be used as either inputs to the LUT-4 or as a Feed-Thru to the CSE via the WLG. The fifth input can be a data port when the LUT is configured as Distributed Memory, a select line for multiplexer operation, or a Feed-Thru directly to the CSE via the WLG (Figure 2).

Look-Up Table – Combinatorial Mode
In combinatorial mode, the LUT-4 can implement any logic function up to four inputs. By using the carry chain and the WLG, each LUT-4 can be combined to form the enhanced functions listed in Table 3.

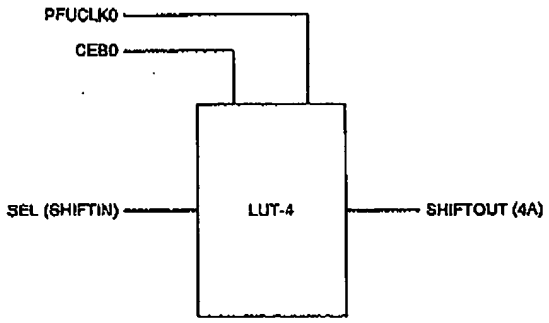
Look-Up Table – Distributed Memory Mode
In the distributed memory mode, the LUT functions as a memory element. The inputs to the LUT function as Address and Data. Each PFU is capable of implementing up to 64 SRAM bits. Both single and double port RAM can be performed in the PFU (Table 3). Furthermore, the distributed memory can be configured as either synchronous or asynchronous memory. Figure 3 illustrates the LUT while in distributed memory mode. When using any LUT in the PFU in memory mode, the Set/Reset signal will be used for Write Enable (WE(SR)) and the CLK0 signal will be used as the clock for synchronous read and write.

Figure 3. LUT in Distributed Memory Mode



Look-Up Table – Shift Register Mode
In the shift register mode, the LUT functions as a 1-bit to 8-bit shift register. This means that each PFU can implement up to four 8-bit shift registers or any cascaded combination. Figure 4 illustrates the LUT when configured in shift register mode.

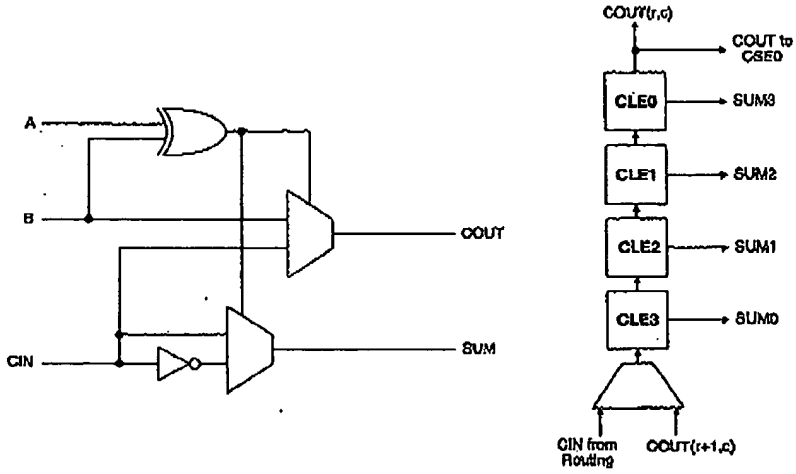
Figure 4. LUT in Shift Register Mode



Carry Chain Generator

The Carry Chain Generator is useful for implementing high-speed arithmetic functions. The CCG consists of a two-input XOR gate whose carryout can be cascaded with the input of the adjacent CCG. As shown in Figure 5, the carryin signal feeds CLE3 of the PFU and is propagated through CLE2 and CLE1 before reaching CLE0. The sum output of the CCG can be fed to the CSE through the WLG. The carryout must propagate to CLE0 for use outside the PFU. The carryout from the PFU can feed the W0 input of CSE0. The CCG also helps to effectively implement wider functions by using its logic elements to expand the capabilities of the LUT-4.

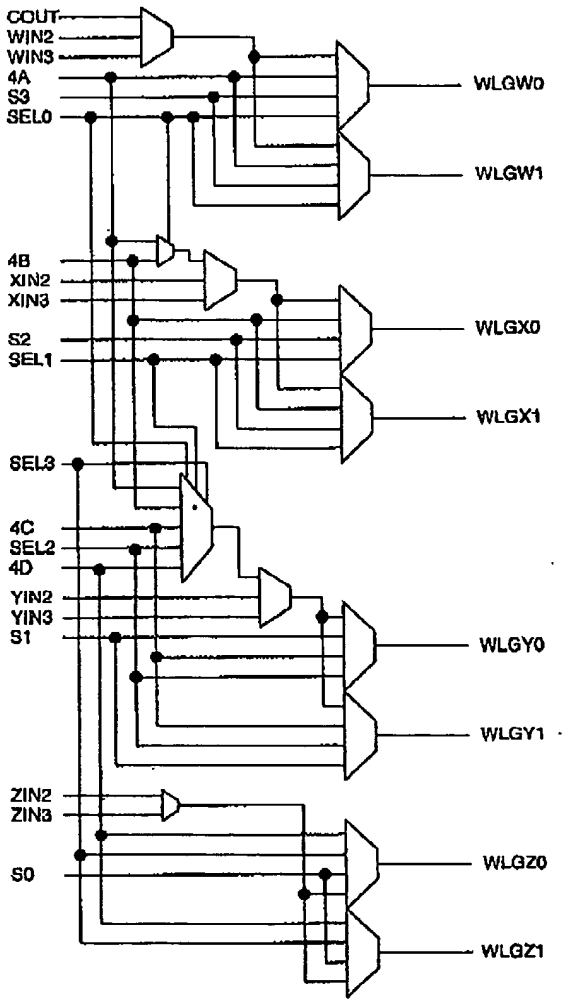
Figure 5. Carry Chain Generator



Wide Logic Generator

The WLG contains the logic necessary to implement wide gate functions. This is made up of a set of multiplexers that are located between the CLE and the CSE. The WLG helps in enhancing the wide gating capability of the PFU. The outputs of each CLE can be cascaded in the WLG to build wide gating functions. Wide multiplexing functions are also possible with a similar use of the WLG. Figure 6 illustrates the WLG.

Figure 6. IspXPGA Wide Logic Generator



Configurable Sequential Element

There are two registers in each CSE for a total of eight registers in each PFU. This high register count assists in implementing efficient pipelined applications with no utilization penalty. Each register can be configured as a latch or D type flip-flop with either synchronous or asynchronous set or reset. Figure 2 shows the signals that feed the register's D inputs. Feed-through signals in the architecture ensure that registers are efficiently utilized even if the accompanying LUT is occupied.

Control Logic

The control signals available to the registers in a PFU are Clock, Clock Enable, and Set/Reset. Figure 7 shows the various options available to generate the clock signal. As can be seen, the clock signal is the output of a 12:1 MUX with true and compliment versions available from the 12:1 MUX. Each CSE can chose whether it uses the true or compliment form of the clock. Figure 8 shows the Set/Reset selection for each PFU in the IspXPGA. A common

Set/Reset signal controls all the registers for each PFU. This common Set/Reset signal is composed of the logical OR term of the Global Set/Reset signal (GSR) and the selected signal from routing. The polarity of this signal is not controllable inside the PFU. Figure 9 shows the Clock Enable and Output Enable selection for each PFU.

Figure 7. Clock Selection per PFU

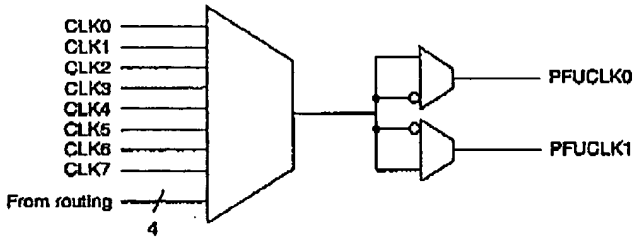


Figure 8. Set/Reset Selection per PFU

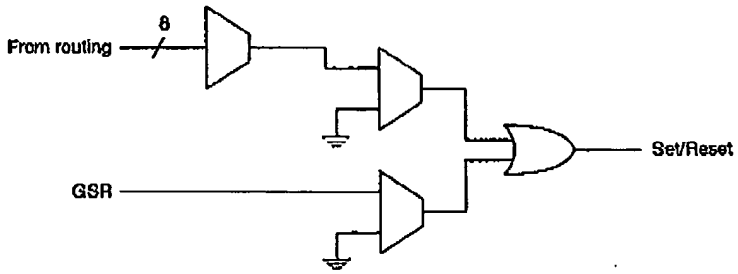
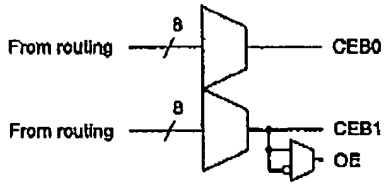


Figure 9. Clock Enable and Output Enable Selection per PFU

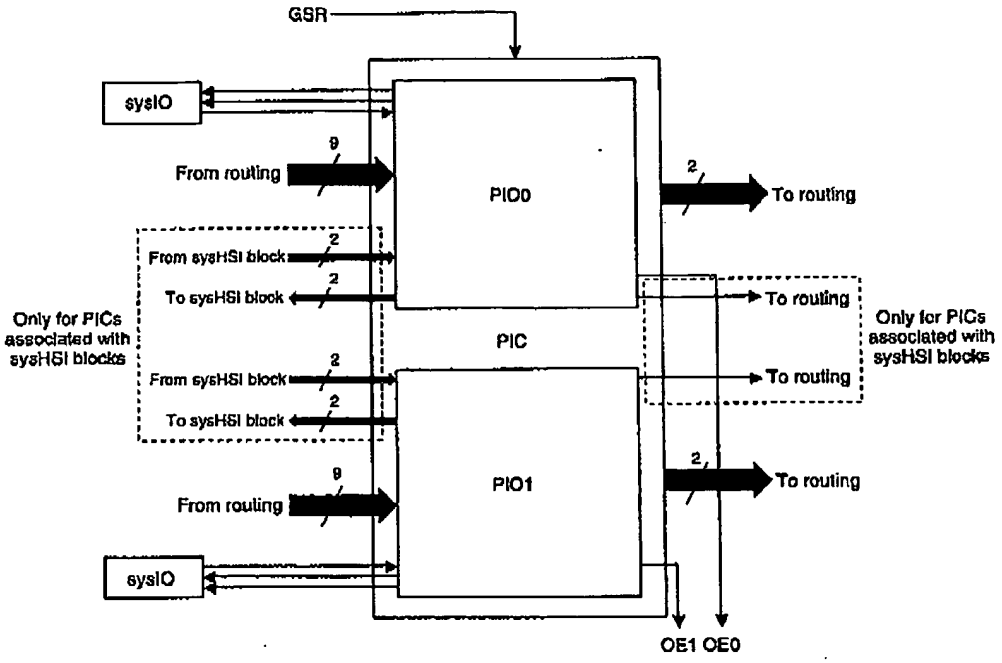


Programmable Input/Output Cell

The Programmable Input/Output Cell (PIC) is an essential part of the symmetrical architecture of the ispXPGA Family. The PICs interface the PFUs and EBRs to the sysIO and sysHSI blocks of the device.

Each PIC contains two Programmable Input/Outputs (PIOs) with a total of 21 inputs and 10 outputs. There are 18 inputs from routing, two inputs from the sysIO buffers, and the Global Set/Reset signal. Four outputs of the PIC connect to routing and two outputs are available as Output Enables for the tri-statable Long Lines. The remaining four outputs feed the sysIO buffers directly (one output enable and one output to each). Each PIC associated with a sysHSI block has four additional inputs and six additional outputs to support the sysHSI blocks. The four additional inputs come from the sysHSI block associated with the PIC. The four of the six additional outputs come from the PIC outputs and feed the sysHSI block, while the remaining two outputs feed routing. Figure 10 shows the block diagram of the PIC with the sysHSI block inputs and outputs.

Figure 10. IspXPGA PIC



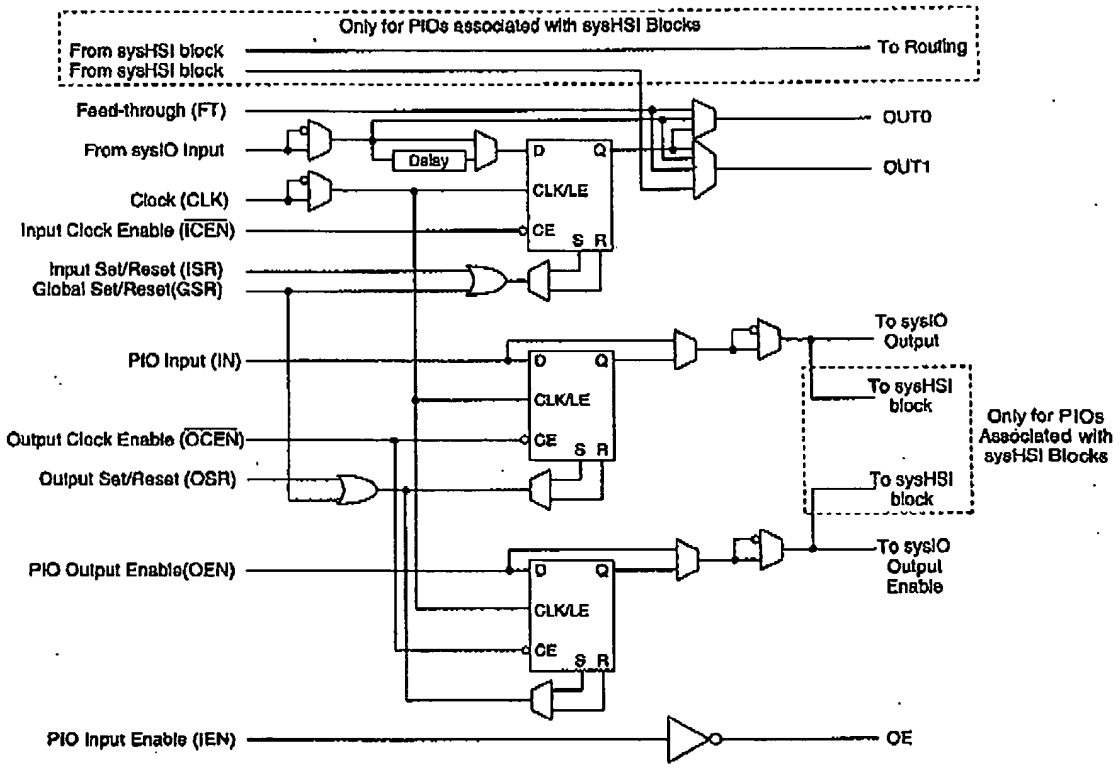
Programmable Input/Output

The PIO is the building block of a PIC. The PIO has a total of 11 inputs and five outputs. Nine of the 11 inputs are generated from routing. The inputs from routing are the PIO Input (IN), Feed-Thru (FT), Clock (CLK), Input Clock Enable (ICE), Input Set/Reset (ISR), Output Clock Enable (OCEN), Output Set/Reset (OSR), PIO Output Enable (OEN), and PIO Input Enable (IEN). The remaining inputs are the sysIO Input buffer signal and the Global Set/Reset signal. Three of the five outputs (OUT0, OUT1, and OE) feed routing. The last two outputs feed the sysIO buffer directly as the output and output enable of the sysIO output buffer.

PIOs associated with sysHSI blocks contain two additional inputs and outputs to support the sysHSI block. The two inputs come from the sysHSI block associated with the PIO, and the two outputs feed the sysHSI block. One of the inputs routes directly through the PIO to routing, while the other is multiplexed with the Feed-Thru, register bypass, and Q output of the register to form the OUT1 output of the PIO. The outputs to the sysHSI block are the same signals as the outputs which feed the sysIO buffers (sysIO Output and sysIO Output Enable).

Each PIO has an input register, an output register, and an output enable register as shown in Figure 11. The input register path of the PIO has a 'delay' option, which slows the data-flow. A two-input OR function of the Global Set/Reset (GSR) and Set/Reset (ISR or OSR) signals creates the set/reset term for the respective registers. Each PIO has two pairs of set/reset and clock enable signals. One is exclusive to the input register, whereas the other is common for both the output and output enable registers. The clock (CLK) is common to all registers in a PIO, and the polarity of the clock is controllable. The input, output, and the output enable registers can be configured as a latch or D-type flip-flop. Each PIO is capable of generating an output enable signal, which in turn becomes a PIC output.

Figure 11. ispXPGA PIO



VLI Routing Resources

The ispXPGA architecture contains a Variable-Length-Interconnect (VLI) routing technology connecting the PFUs, PICs, and EBRs in the device. There are four types of routing resources, Global Lines, Long Lines, General Interconnect, and Local Lines forming the global routing structure. This allows a signal to be routed to any element in the device with the optimal delay.

The Global Lines consist of global clock lines and a global set/reset line. These lines are routed to all elements in the device. They are specifically designed for high speed, predictable timing regardless of fan-out. The global clock lines can also be used as dedicated inputs.

The Long Lines consist of Horizontal and Vertical Long Lines (HLL and VLL). The VLL and HLL are tri-statable lines spanning the entire device. These lines allow fast routing for high fan-out nets and general-purpose functions.

The General Interconnect consists of Double and Deca Lines. The Double Lines connect up to three elements (two plus the driving element), while the Deca Lines connect up to eleven elements (ten plus the driving element).

The Local Lines are extremely fast routing paths consisting of Feedback and Direct Connect Lines. The Feedback Lines are internal routing paths from the PFU outputs to the PFU inputs. The Direct Connect Lines connect all adjacent elements.

The Common Interface Block (CIB) provides the link between the logic element (PFU, PIC, or EBR) and the VLI Routing resources. The CIB is a switch matrix that can be programmed to connect virtually any routing resource to any input or output of the logic element.

Memory

The ispXPGA architecture provides a large amount of resources for memory intensive applications. Embedded Block RAMs (EBRs) are available to complement the Distributed Memory that is configured in the PFUs (see Look-Up Table -Distributed Memory Mode in the PFU section above). Each memory element can be configured as RAM or ROM. Additionally, the internal logic of the device can be used to configure the memory elements as FIFO and other storage types. These EBRs are referred to as sysMEM blocks. Refer to Table 1 for memory resources per device.

sysMEM Blocks

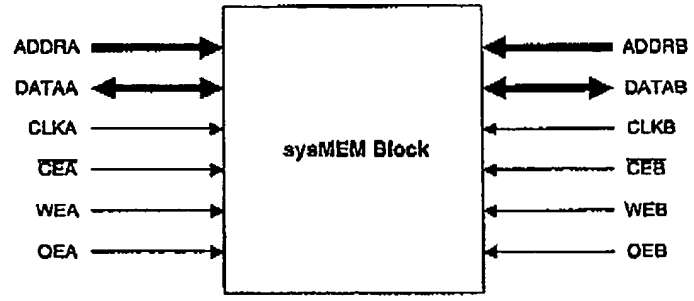
The sysMEM blocks are organized in columns distributed throughout the device. Each EBR contains 4.6K bits of dual-port RAM with dedicated control, address, and data lines for each port. Each column of sysMEM blocks has dedicated address and control lines that can be used by each block separately or cascaded to form larger memory elements. The memory cells are symmetrical and contain two sets of identical control signals. Each port has a read/write clock, clock enable, write enable, and output enable. Figure 12 illustrates the sysMEM block.

The ispXPGA memory block can operate as single-port or dual-port RAM. Supported configurations are:

- 512 x 9 bits single-port (8 bits data / 1 bit parity)
- 256 x 18 bits single-port (16 bits data / 2 bits parity)
- 512 x 9 bits dual-port (8 bits data / 1 bit parity)
- 256 x18 bits dual-port (16 bits data / 2 bits parity)

The data widths of "9" and "18" are ideal for applications where parity is necessary. This allows 9 data bits, 8 data bits plus a parity bit, 18 data bits, or 16 data bits plus two parity bits. The logic for generating and checking the parity must be customized separately.

Figure 12. sysMEM Block Diagram

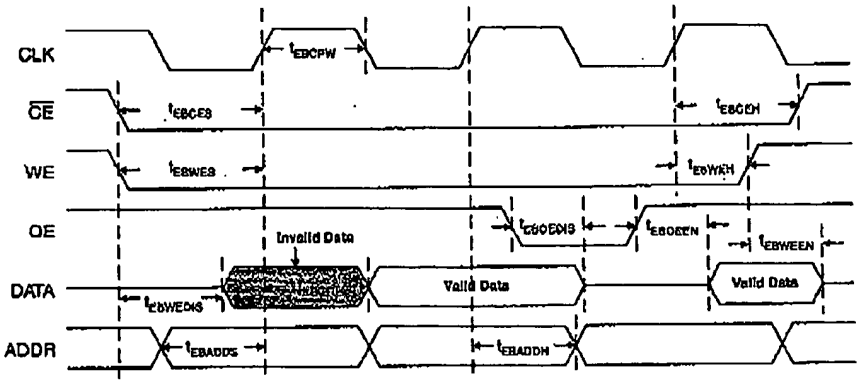


Read and Write Operations

The ispXPGA EBR has fully synchronous read and write operations as well as an asynchronous read operation. These operations allow several different types of memory to be implemented in the device.

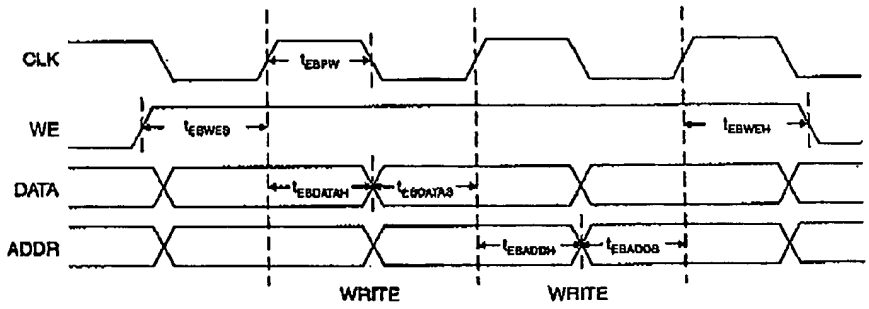
Synchronous Read: The Clock Enable ($\overline{CE}$) and Write Enable (WE) signals control the synchronous read operation. When the $\overline{CE}$ signal is low, the clock is enabled. When the WE signal is low the read operation begins. Once the address (ADDR) is present, a rising clock edge (or falling edge depending on polarity) causes the stored data to be available on the DATA port. Figure 13 illustrates the synchronous read timing.

Figure 13. EBR Synchronous Read Timing Diagram



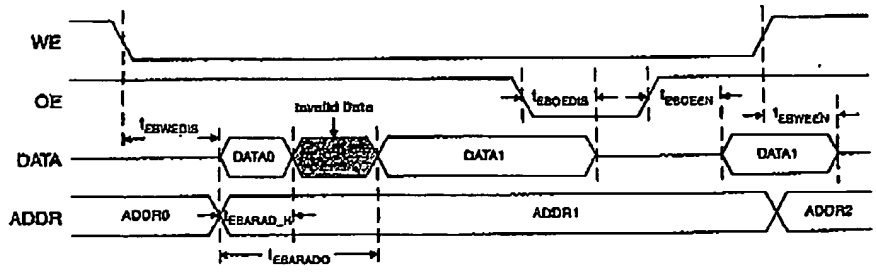
Synchronous Write: The WE signal controls the synchronous write operation. When the WE signal is high and the write operation begins. Once the address and data are present and the Output Enable (OE) is active, a rising clock edge (or falling edge depending on polarity) causes the data to be stored into the EBR. Figure 14 illustrates the synchronous write timing.

Figure 14. EBR Synchronous Write Timing Diagram



Asynchronous Read: The WE signal controls the asynchronous read operation. When the WE signal is low, the read operation begins. Shortly after the address is present, the stored data is available on the DATA port. Figure 15 illustrates the asynchronous read timing. For more information about the EBR, refer to Lattice technical note number TN1028 *IspXPGA Memory Usage Guidelines*, available at www.latticesemi.com.

Figure 15. EBR Asynchronous Read Timing Diagram



sysCLOCK PLL Description

The sysCLOCK PLL circuitry consists of Phase-Lock Loops (PLLs) and the various dividers, reset, and feedback signals associated with the PLLs. This feature gives the user the ability to synthesize clock frequencies and generate multiple clock signals for routing within the device. Furthermore, it can generate clock signals that are aligned either at the board level or the device level.

The ispXPGA devices provide up to eight PLLs. Each PLL receives its input clock from its associated global clock pin, and its output is routed to the associated global clock net. For example, PLL0 receives its clock input from the GCLK0 global clock pin and provides output to the CLK0 global clock net. The PLL also has the ability to output a secondary clock that is a division of the primary clock output. When using the secondary clock, the secondary clock will be routed to the neighboring global clock net. For example, PLL0 will drive its primary clock output on the CLK0 global clock net and its secondary clock output will drive the CLK1 global clock net. Additionally, each PLL has a set of PLL_RST, PLL_FBK, and PLL_LOCK signals. The PLL_RST signal can be generated through routing or a dedicated dual-function I/O pin. The PLL_FBK signal can be generated through a dedicated dual-function I/O pin or internally from the Global Clock net associated with the PLL. The PLL_LOCK signal feeds routing directly from the sysCLOCK PLL circuit. Figure 17 illustrates how the PLL_RST and PLL_FBK signals are generated.

Each PLL has four dividers associated with it, M, N, V, and K. The M divider is used to divide the clock signal, while the N divider is used to multiply the clock signal. The V divider allows the VCO frequency to operate at higher frequencies than the clock output, thereby increasing the frequency range. The K divider is only used when a secondary clock output is needed. This divider divides the primary clock output and feeds to the adjacent global clock net. Different combinations of these dividers allow the user to synthesize clock frequencies. Figure 16 shows the ispXPGA PLL block diagram.

The PLL also has a delay feature that allows the output clock to be advanced or delayed to improve set-up and clock-to-out times for better performance. This operates by inserting delay on the input or feedback lines of the PLL. For more information on the PLL, please refer to Lattice technical note number TN1003, *sysCLOCK PLL Usage and Design Guidelines*, available at www.latticesemi.com.

Figure 16. ispXPGA PLL Block Diagram

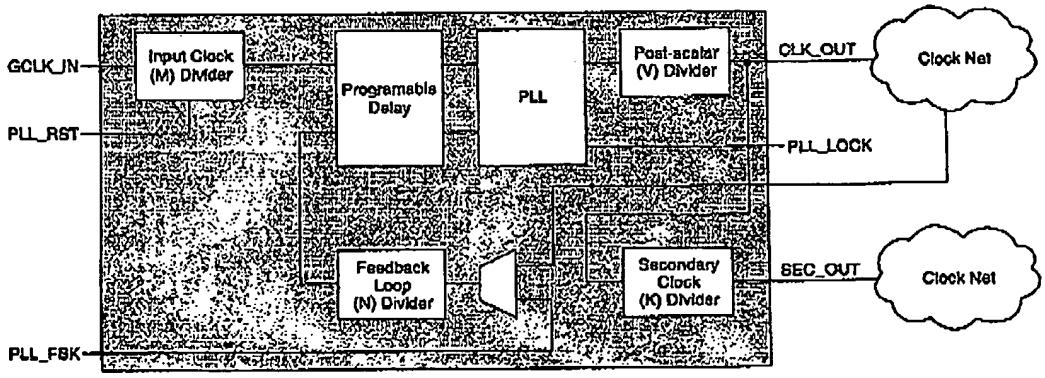
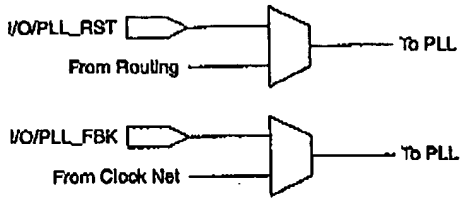


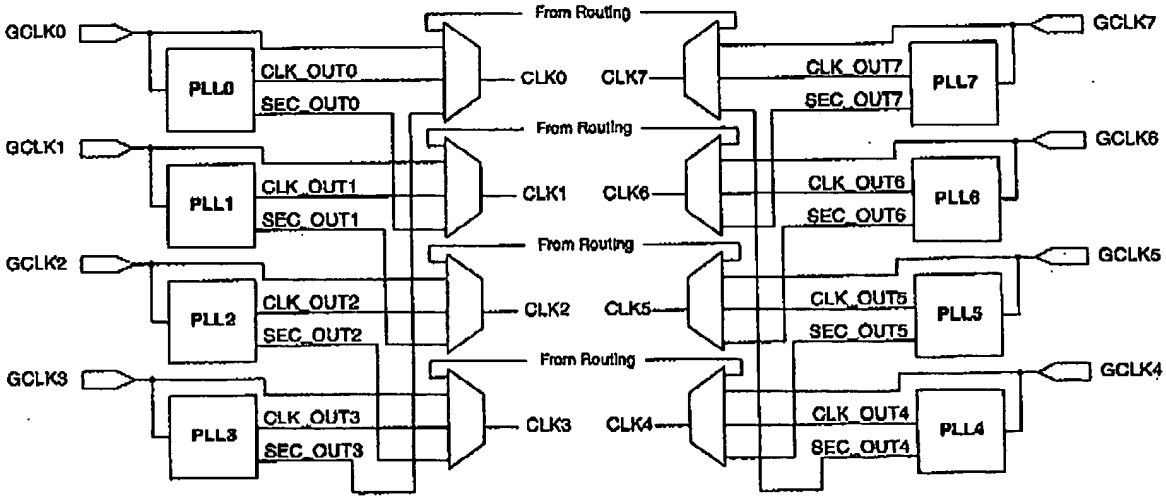
Figure 17. ispXPGA PLL_RST and PLL_FBK Generation



Clock Routing

The Global Clock Lines (GCLK) have two sources, their dedicated pins and the sysCLOCK circuit. Figure 18 illustrates the generation of the Global Clock Lines.

Figure 18. Global Clock Line Generation



sysIO Capability

All the ispXPGA devices have eight sysIO banks, where each bank is capable of supporting multiple I/O standards. Each sysIO bank has its own I/O supply voltage (V_{CCO}) and reference voltage (V_{REF}) resources allowing each bank complete independence from the others. Each I/O is individually configurable based on the bank's V_{CCO} and V_{REF} settings. In addition, each I/O has configurable drive strength, weak pull-up, weak pull-down, or a bus-keeper latch. Table 4 lists the number of I/Os supported per bank in each of the ispXPGA devices.

Table 5 lists the sysIO standards with the typical values for V_{CCO}, V_{REF} and V_{TT}.

The TOE, JTAG TAP pins, PROGRAM, CFG0 and DONE pins of the ispXPGA device are the only pins that do not have the sysIO capabilities. The TOE and CFG0 pins operate off the V_{CC} of the device, supporting only the LVCMOS standard corresponding to the device supply voltage. The TAP pins have a separate supply voltage (V_{CCJ}), which determines the LVCMOS standard corresponding to that supply voltage.

There are three classes of I/O interface standards that are implemented in the ispXPGA devices. The first is the unterminated, single-ended interface. It includes the 3.3V LVTTL standard along with the 1.8V, 2.5V, and 3.3V LVCMOS interface standards. Additionally, PCI and AGP-1X are subsets of this type of interface.

The second type of interface implemented is the terminated, single-ended interface standard. This group of interfaces includes different versions of SSTL and HSTL interfaces along with CTT, and GTL+. Usage of these particular I/O interfaces requires an additional V_{REF} signal. At the system level a termination voltage, V_{TT}, is also required. Typically an output will be terminated to V_{TT} at the receiving end of the transmission line it is driving.

The third type of interface standards are the differential standards LVDS, BLVDS, and LVPECL. The differential standards require two I/O pins to create the differential pair. The logic level is determined by the difference in the two signals. Table 6 lists how these interface standards are implemented in the ispXPGA devices.

For more information on sysIO capability, refer to Lattice technical note number TN1000, *sysIO Usage Guidelines for Lattice Devices* available at www.latticesemi.com.

Figure 19. sysIO Banks per Device

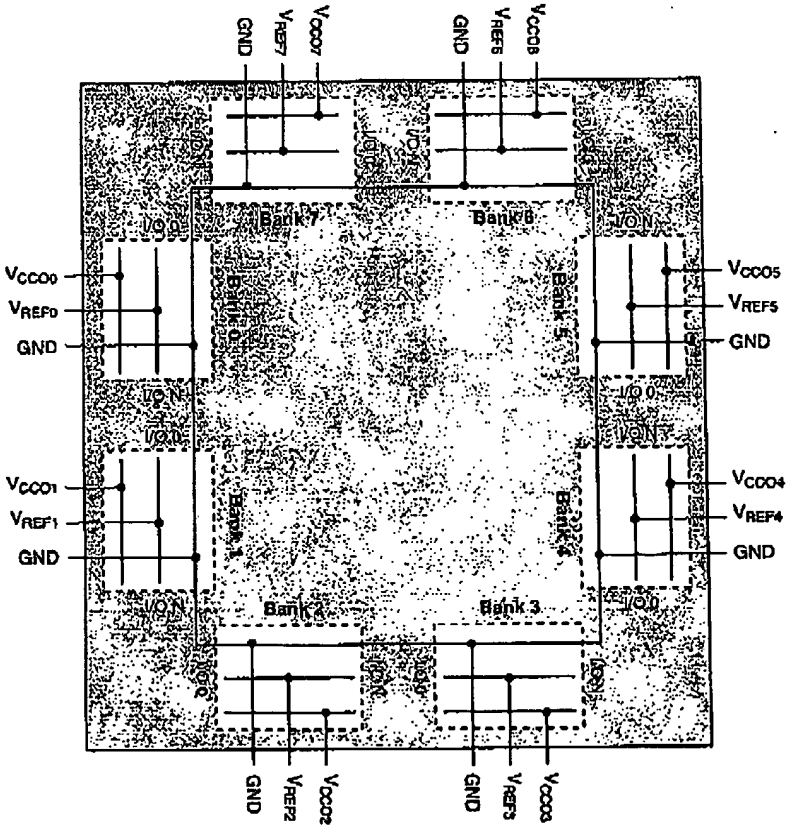


Table 4. Number of I/Os per Bank

Device	Max. Number of I/Os per Bank (N)
XPGA 1200	62
XPGA 500	42
XPGA 200	26
XPGA 125	22

Table 5. IspXPGA Supported I/O Standards

sysIO Standard	V_{CCO}	V_{REF}	V_{TT}
LVTTL	3.3V	N/A	N/A
LVCMOS-3.3	3.3V	N/A	N/A
LVCMOS-2.5	2.5V	N/A	N/A
LVCMOS-1.8	1.8V	N/A	N/A
PCI	3.3V	N/A	N/A
AGP-1X	3.3V	N/A	N/A
SSTL3, Class I, II	3.3V	1.5V	1.5V
SSTL2, Class I, II	2.5V	1.25V	1.25V
HSTL, Class I	1.5V	0.75V	0.75V
HSTL, Class III	1.5V	0.9V	1.5V
GTL+	N/A	1.0V	1.5V
LVPECL	3.3V	N/A	N/A
LVDS[1]	2.5V	N/A	N/A
BLVDS	2.5V	N/A	N/A

1. V_{CCO} must be 2.5V for high speed serial operations (sysHSI block).

Table 6. Differential Interface Standard Support[1]

		sysIO Buffer Not Using sysHSI Block	sysIO Buffer Using sysHSI Block
LVDS	Driver	Supported with external resistor network	Supported
	Receiver	Supported with standard termination	Supported with standard termination
BLVDS	Driver	Supported with external resistor network	Not supported
	Receiver	Supported (may need termination)	Supported (may need termination)
LVPECL	Driver	Supported with external resistor network	Not supported
	Receiver	Supported with termination	Supported with termination

1. For more information, refer to Lattice technical note TN1000, *sysIO Usage Guidelines*, available at www.latticesemi.com.

High Speed Serial Interface Block (sysHSI Block)

The High Speed Serial Interface (sysHSI) allows high speed serial data transfer over a pair of LVDS I/O. The IspXPGA devices have multiple sysHSI blocks.

Each sysHSI block has two SERDES blocks which contain two main sub-blocks, Transmitter (with a serializer) and Receiver (with a deserializer) including Clock/Data Recovery Circuit (CDR). Each SERDES can be used as a full duplex channel. The two SERDES in sysHSI blocks share a common clock and must operate at the same nominal frequency. Figure 20 shows the sysHSI block.

Device features support two data coding modes: 10B/12B and 8B/10B (for use with other encoding schemes, see Lattice's sysHSI technical notes). The encoding and decoding of the 10B/12B standard are performed within the sysHSI block. For the 8B/10B standard, the symbol boundaries are aligned internally but the encoding and decoding are performed outside the sysHSI block.

Each SERDES block receives a single high speed serial data input stream (with embedded clock) from an input, and provide a low speed 10-bit wide data stream and a recovered clock to the device. For transmitting, SERDES converts a 10-bit wide low-speed data stream to a single high-speed data stream with embedded clock for output.

Additionally, multiple sysHSI blocks can be grouped together to form a source synchronous interface of 1-10 channels.

Table 7 shows the clock sources available for the REFCLKs of the different sysHSI blocks. The Signal Description table in this data sheet provides the descriptions of the sysHSI block inputs and outputs.

For more information on the SERDES/CDR, refer to Lattice technical note number TN1020, *sysHSI Usage Guidelines.*

Figure 20. sysHSI Block Diagram

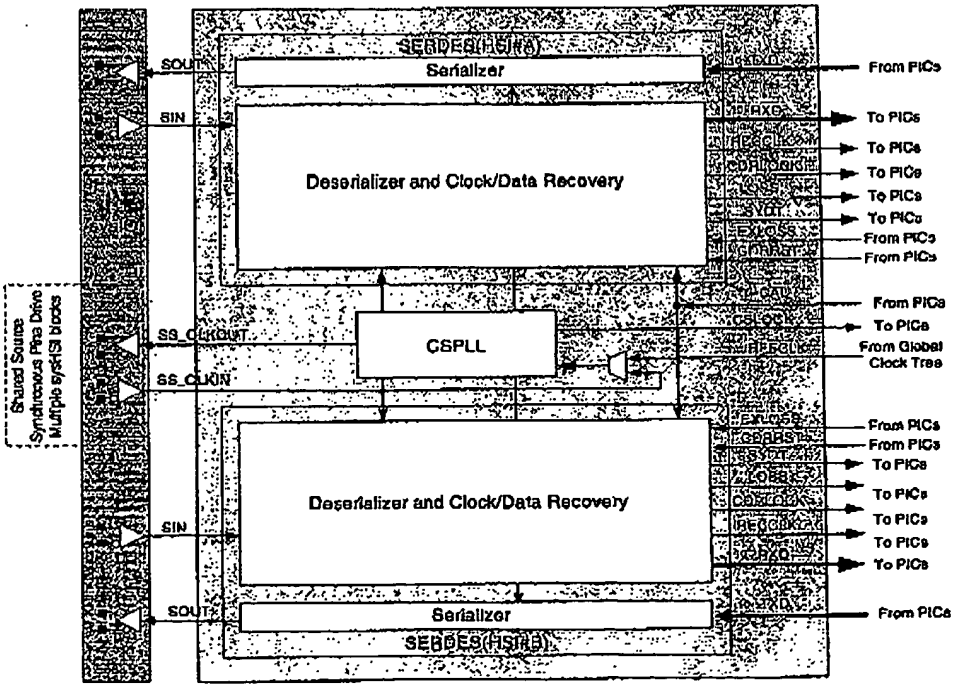


Table 7. sysHSI Block REFCLK Selections[1]

sysHSI Block	Available Global Clock Nets
0	CLK0, CLK1, CLK2, CLK3
1	CLK0, CLK1, CLK2, CLK4
2	CLK0, CLK1, CLK2, CLK5
3	CLK0, CLK1, CLK3, CLK6
4	CLK0, CLK1, CLK3, CLK7
5	CLK0, CLK3, CLK5, CLK7
6	CLK0, CLK2, CLK5, CLK7
7	CLK0, CLK1, CLK5, CLK6
8	CLK0, CLK5, CLK6
9	CLK0, CLK5, CLK6, CLK7

1. Table 6 applies to all devices. Ignore sysHSI blocks not available in a specific device.

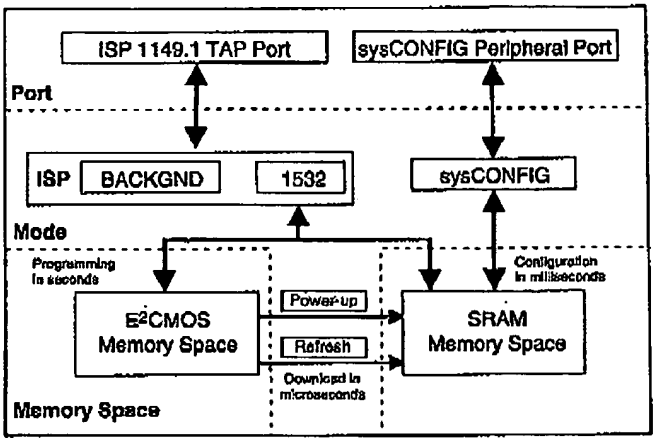
Configuration and Programming

The IspXPGA family of devices takes a unique approach to FPGA configuration memory. It contains two types of memory, Static RAM and non-volatile E²CMOS cells. The static RAM is used to control the functionality of the device during normal operation and the E²CMOS memory cells are used to load the SRAM. The E²CMOS memory module can be thought of as the hard drive for the IspXPGA configuration and the SRAM as the working configuration memory. There is a one-to-one relationship between SRAM memory and the E²CMOS cells. The SRAM can be configured either from the E²CMOS memory or from an external source, as shown in Figure 21.

Figure 21 shows the different ports and modes that are used in the configuration and programming of the IspXPGA devices. There are two possible ports that can be used for configuration of the SRAM memory: the ISP port which is compliant to the IEEE 1149.1 Test Access Port (TAP) Std. and the ISP port which accommodates bit-wide configuration. The sysCONFIG port allows byte-wide configuration of the SRAM configuration memory. When programming the E²CMOS memory, only the 1149.1 TAP can be used.

Configuration and programming done through the 1149.1 Test Access Port (TAP) are fully compliant to both the IEEE Std. 1149.1 Boundary Scan TAP specification and the IEEE Std. 1532 In-System Configuration specification. To configure or program the device using the 1149.1 TAP the device must be in the ISP mode. To configure the SRAM memory using the sysCONFIG Port, the device must be in the sysCONFIG mode. Upon power-up, the device's SRAM memory can be configured either from the E²CMOS memory or from an external source through the sysCONFIG mode. Additionally, the SRAM can be re-configured from the E²CMOS memory by executing a "REFRESH." See Lattice technical note number TN1026, *ispXP Configuration Usage Guide,* for more in depth information on the different programming modes, timing and wake-up, available at www.latticesemi.com.

Figure 21. ispXP Block Diagram



IEEE 1149.1-Compliant Boundary Scan Testability

All ispXPGA devices have boundary scan cells and are compliant with the IEEE 1149.1 standard. This allows functional testing of the circuit board on which the device is mounted through a serial scan path that can access all critical logic notes. Internal boundary scan registers are linked internally, allowing test data to be shifted in and loaded directly onto test nodes, or test node data to be captured and shifted out for verification. In addition, these devices can be linked into a board-level serial scan path for more board level testing.

Security Scheme

A programmable security scheme is provided on the ispXPGA devices as a deterrent to unauthorized copying of the array configuration patterns. Once programmed, the security scheme prevents read-back of the programmed pattern by a device programmer, securing proprietary designs from competitors. The entire device must be erased in order to erase the security scheme.

Density Shifting

The ispXPGA family has been designed to ensure that different density devices in the same package have the same pin-out. Furthermore, the architecture ensures a high success rate when performing design migration from lower density parts to higher density parts. In many cases, it is possible to shift a lower utilization design targeted for a high-density device to a lower density device. However, the exact details of the final resource utilization will impact the likely success in each case.